REMARKS/ARGUMENTS

These remarks and amendments are filed in response to the Office Action mailed November 12, 2009. As this response is timely filed within the 3-month shortened statutory period, no fee is believed due. Nonetheless, the Examiner is expressly authorized to charge any deficiencies to Deposit Account No. 50-0951.

Claims 1, 4, 7, 9, 11, 13, 14, 17, 19, 22, 23, 26, 28, 31, 32, 35 and 37 are rejected under 35 U.S.C. 103(a) as being obvious over Blaukopf in view of Sun. Claims 1, 4, 7, 9 and 11 include the feature of the module including a datagram packet application programming interface configured to enable an application developer to specify a length of fragmentation and specify routing of fragmented data to a destination address and port number. The Office Action concedes that Blaukopf does not disclose this feature but asserts that it would have been obvious to modify the Blaukopf system with this claimed feature based on the teachings of Sun because "the developer specifies the protocol with which the mediate servlet is to communicate with remote service." Applicants respectfully disagree and assert that Blaukopf teaches away from the Office Action's suggested modification.

Blaukopf is directed towards a system of code optimization which is intended to circumvent the perceived problems of manual code adjustment being cumbersome and labor intensive. (Blaukopf par. 13). To this end, Blaukopf provides a code generator which can load code templates for generating the desired code:

> The code generator 42 loads one or more of the code templates 56 in preparation for generating its output. Each of the code templates 56 is a blueprint for the code to be generated. It contains two interleaved parts, data 66, and logic 68. The data 66 consists of Java source code, which is to be copied verbatim to output files. The logic 68 is typically Java byte code, (e.g., J2ME proxy classes and their infrastructure) that is executed during code generation to control what parts of the data 66 are to be copied to the mediator

> servlet 22. For example, the logic 68 in one of the code templates 56 may ensure that data 66 that contains code to support grouped calls, or to handle receiving arrays of strings from the server, is only included if there is an explicit user requirement for such functionality, or an implicit requirement that can be inferred from the configuration data. By appropriately configuring the data 66 and the logic 68, the code templates 56 relieves the developer 24 (FIG. 1) of much of the burden of providing detailed code. Code templates can be provided for many different products that can function as the mobile information device 12 (FIG. 1), Similarly, different code templates can be prepared for different web services. As the clients and web services encountered today are so diverse, it is likely that large catalogs of code templates 56 will be maintained for use by the code generator 42. The output of the code generator 42, and the optimized protocol 46 (FIG. 2) is influenced by the input data 62, in which the developer 24 (FIG. 1) specifies the protocol with which the mediator servlet 22 is to communicate with the remote service. As mentioned above, the developer 24 has wide latitude to select a standard or a non-standard protocol. (Blaukopf par. 98).

While it is true that the developer is given the opportunity in Blaukopf to select a protocol, Blaukopf makes clear that the ultimate optimized protocol is generated by the code generator 42. (Blaukopf par. 98). Blaukopf makes clear that it is the use of this automatically generated "optimized protocol" which allows the client code to be minimized:

> In some aspects of the present invention the problem of client code optimization is solved by a technique, wherein a developer prepares an application for a resource-constrained client, such as a mobile information device, in which he "specifies services to be exported to the client from server-side providers. A development tool incorporating a template-driven code generator is invoked to automatically create specialized client and server code, optimized for the specified services. **Use of an optimized protocol allows client code to be minimized through elimination of redundant code, and by using code optimizations that can be automatically selected by the code generator.** The server code consists of a mediator that acts as a gateway between the client and providers of the specified services, translating the optimized

> protocol into a standard protocol that is expected by the providers. The server-side providers thus see a conventional client, and need not modify their standard procedures in any way in order to receive requests from such low-end clients. (Blaukopf par. 16)(emphasis added).

One of ordinary skill in the art would not modify the Blaukopf system to include a module having a datagram packet application programming interface configured to enable an application developer to specify a length of fragmentation and specify routing of fragmented data to a destination address and port number. This is particularly so where Blaukopf provides for its code generator to have a great degree of autonomy in achieving a perceived optimization, such as through automatically making selections as to different implementations of a method and automatically determining the distribution of computation:

> In some embodiments, the code generator is aware of the characteristics of the client, which knowledge is exploited for even greater code optimization. For example, the code generator may also select from among different implementations of a method in order to produce a minimal code size. The results are superior to those that could be obtained by an optimizing compiler or code obfuscation engine.
> An advantage of some aspects of the present invention is minimization of the computational load on the client by automatically and optimally distributing the computation between the client and the server. (Blaukopf paragraphs 17 and 18).

Claims 13, 14, 17 and 19 include the step of providing a datagram packet application programming interface configured to enable an application developer to specify with the generated program code a length of fragmentation of a data string and to specify routing of fragmented data to a destination address and port number. For the reasons described above, Blaukopf teaches away from this feature.

Claims 22, 23, 26 and 28 include the features of a datagram packet application programming interface configured to enable an application developer to specify with the generated program code a length of fragmentation of a data string and to specify routing of fragmented data to a destination address and port number. For the reasons described above, Blaukopf teaches away from this feature.

Claims 31, 32, 35 and 37 include the features of generating and providing to a system user a datagram packet application programming interface configured to enable an application developer to specify with the generated program code a length of fragmentation of a data string and to specify routing of fragmented data to a destination address and port number. For the reasons described above, Blaukopf teaches away from this feature.

Claims 2, 3, 10, 20, 29, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blaukopf in view of Sun and further in view of Wahli. Claims 2, 3, 10, 20, 29, and 38 depend from claims 1, 13, 22 and 31, respectively. For the reasons described above, claims 2, 3, 10, 20, 29, and 38 are patentable over the cited combination of art.

Claims 5, 6, 8, 12, 15, 16, 18, 21, 24, 25, 27, 30, 33, 34, 36 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blaukopf in view of Sun and further in view of Fox. Claims 5, 6, 8, 12, 15, 16, 18, 21, 24, 25, 27, 30, 33, 34, 36 and 39 depend from claims 1, 13, 22 and 31, respectively. For the reasons described above, claims 5, 6, 8, 12, 15, 16, 18, 21, 24, 25, 27, 30, 33, 34, 36 and 39 are patentable over the cited combination of art.

CONCLUSION

Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

AKERMAN SENTERFITT

Date: February 12, 2009

/ANDREW C. GUST/

Andrew C. Gust.
Reg. No. 47,620
AKERMAN SENTERFITT
P.O. Box 3188
West Palm Beach, FL 33402-3188
Tel: 561-653-5000

Docket No. 5853-302-1